

**SYSTEM, METHOD AND APPARATUS FOR MONITORING
AND EXECUTION OF ENTRY AND EXIT ORDERS**

Inventor(s): William R. Cruz
Ralph L. Cruz
Peter A. Parandjuk

Continuation of Application

Serial No. 09/968,849

Design Application

Plant Application

PCT National Stage

PCT International Stage

Date Filed:

X

1/4/02

SYSTEM, METHOD AND APPARATUS FOR MONITORING AND EXECUTION OF ENTRY AND EXIT ORDERS

CROSS REFERENCE TO PRIORITY APPLICATION

The present application claims priority on a provisional application, U.S.
5 Serial No. 60/237,870, entitled "System, Method and Apparatus for Monitoring and
Execution of Buy and Sell Orders", filed on October 4, 2000, and a nonprovisional
application, U.S. Serial No. 09/968,849, entitled "System, Method and Apparatus for
Monitoring and Execution of Entry and Exit Orders", filed on October 3, 2001, which
are incorporated herein by reference.

BACKGROUND OF THE INVENTION

Field of the Invention

The present invention relates generally to methods for trading securities,
stocks, options, indexes, commodities and futures in a distributed network
environment. In particular, the invention relates to a method and system in which a
15 trader can write a trading strategy, back test the trading strategy and then automate
that trading strategy to automatically generate entry and exit orders based upon the
strategy. The invention constantly monitors market and strategy conditions and
intelligently routes entry and exit orders for execution based upon the trading strategy
and the monitored conditions. In addition, the invention alerts the user of trading
20 opportunities due to changing market conditions and further informs the user of the
status of the generated entry and exit orders.

BACKGROUND AND OBJECTS OF THE INVENTION

Previous art related to Internet-based trading allows individual traders,
professional traders and institutional traders to trade securities based upon market

data. The existing prior art, however, does not allow traders to write trading strategies and back test the strategies on historical data prior to automatically implementing the strategies into real markets.

The financial industry has seen dramatic change over the past decade. The
5 development of new technologies that quickly deploy information over the Internet
has changed the methods of trading securities and commodities. The Internet has
made it possible for institutional and other professional traders to improve their
decisions relating to the management, placement, and execution of their trades. The
Internet has also made it possible for individual traders to directly manage their own
10 portfolios from their home. Many traders, who at one time relied entirely upon the
assistance of a stock broker or other securities professional now independently
manage their own portfolios.

A trader may notice reoccurring patterns in a market and place trades that
capitalize on those patterns. By way of example, a trader may notice that every time
15 the market closes up after a three-day low, the market will continue to climb. A
sophisticated trader may view this occurrence as a buy opportunity to enter the
market. In addition, a sophisticated trader will likely have an exit strategy for that
particular trade. Various trading strategies may be based on market momentum,
price, volume or time, or some combination thereof.

20 Traders develop trading strategies based on historical market data, experience
and occurrences. Prior to the Internet, market data was available to traders mainly
through satellite feeds, print media or television. Instant market data is now widely
available from a variety of sources, including various web sites and on-line brokerage

5 firms. Traders also develop trading strategies by recognizing patterns in the market. For example, if a trader recognizes that a particular security continually bottoms out at a price of \$50 and routinely bounces back to a higher price, that trader will develop a trading strategy that will focus on buying that particular security at its bottom price and selling it at its peak price. Many trading strategies are mathematical in nature and depend on a variety of variables. Since trading strategies are implemented through a variety of variables, they are readily described by the use of algorithms. Therefore, trading strategies may be automated through the use of computer programs.

10 Currently, there does not exist a means to automate the process of back testing a trading strategy against historical market data and then apply that trading strategy to real time market data to automatically generate entry and exit orders with the changing conditions of real-time markets. By applying the rules of a new trading strategy to historical data through a computerized process, a computer can show the historical success of that strategy on the data in question. The trader can then better determine the likelihood of success of that strategy when it will be applied to real-time market trading. As a result, the trader can minimize the risk associated with the trade. Further, by applying a trading strategy to real time market data to automatically generate entry and exit orders with the changing conditions of real-time markets, a trader minimizes the risk of missing a trading opportunity for a particular strategy. 15 Thus, it would be an improvement over the prior art to automate back testing and the application of trading strategies through the use of computers. 20

There is, therefore, a present need to provide an improved system and method for a trader to develop a trading strategy, back test the trading strategy on real market

data and to automate the execution of the trading strategy with the changing conditions of real-time markets.

It is, accordingly, an object of the present invention to set forth an improved system and method for automating the monitoring and execution of trading strategies.

5 It is a further object of the present invention to generate entry and exit orders based upon the users' trading strategies and real-time market data.

It is further object of the present invention to provide the user with the ability to write trading strategies based upon market conditions and trends.

10 It is further object of the present invention to provide the user with the ability to write trading strategies in computer code for the automated execution of the trading strategy.

It is further object of the present invention to provide the user with the ability to write trading strategies and rules in English-like phrases and codes that are interpreted by a computer into an executable trading strategy.

15 It is a further object of the present invention to provide the user with market data and information to use in developing trading strategies.

It is further object of the present invention to automate the process of back testing trading strategies on real, historical market data.

20 It is a further object of the present invention to perform initial checks on generated entry and exit orders to determine if the order is restricted from being filled because of time, price or position restrictions.

It is a further object of the present invention for the computer coded trading strategies to be used in conjunction with real time market data to automate the placing of orders in accordance with the trading strategy.

In accordance with another aspect of the invention, it is a further object of the present invention to monitor the trading price of securities for which a user has an active order.

It is a further object of the present invention to automatically fill entry and exit orders based upon price restrictions when those restrictions are fulfilled.

In accordance with a further aspect of the invention it is a further object of the invention to constantly check the status of entry and exit orders on a pre-determined time interval.

It is a further object of the present invention to alert the user of the invention if the entry and exit orders have not been filled after a predetermined time.

In accordance with a further aspect of the invention it is a further object of the invention to automatically fill entry and exit orders as soon as market conditions change that make the orders become unrestricted.

In accordance with another aspect of the invention, it is a further object of the present invention to monitor market conditions affecting the users' trading strategies.

It is a further object of the invention to alert the user if an entry or exit order was not filled and the market conditions that triggered the order no longer exist.

In accordance with another aspect of the invention, it is a further object of the present invention to monitor the users' active trading strategies.

It is a further object of the invention to alert the user if an entry or exit order was not filled and the trading strategy that triggered the order is no longer active.

Finally, and in accordance with an additional aspect of the invention, it is a further object of the invention to modify entry and exit orders in accordance with the
5 changing market conditions.

SUMMARY OF THE INVENTION

The present invention discloses a method and system to create and write a trading strategy, back test the strategy and automatically execute and monitor the strategy in an Internet-based trading environment. The automated process
10 automatically generates entry and exit orders and places those trades. In contrast to traditional methods, the present invention involves a method and system in which the user can write a computer executable trading strategy in English-like phrases and codes. In addition, the invention allows the user to back test the trading strategy on real, historical market data prior to implementing the strategy for automatic execution.

15 A more complete appreciation of the present invention and the scope thereof can be obtained from the accompanying drawings which are briefly summarized below in the following detailed description of the presently-preferred embodiments of the invention and the appended claims.

BRIEF DESCRIPTION OF THE DRAWINGS

20 The objects, features and advantages of the present invention are readily apparent from the detailed description of the preferred embodiments set forth below, in conjunction with the accompanying Drawings in which:

FIGURE 1 is a diagram illustrating a network environment utilizing the principles of the present invention;

FIGURE 2 is a high-level flowchart depicting the automation process for strategy entry orders of the present invention;

5 FIGURE 3 is a flow chart that depicts the process and methodology of the Initial Checks module for entry orders;

FIGURE 4 is a flow chart that depicts the process and methodology of the Price Event from Data Server module for entry orders;

10 FIGURE 5 is a flow chart that depicts the process and methodology of the Place Trade Event module for entry orders;

FIGURE 6 is a flow chart that depicts the process and methodology of the Timer Event module for entry orders;

FIGURE 7 is a flow chart that depicts the process and methodology of the Trade Server Event module for entry orders;

15 FIGURE 8 is a flow chart that depicts the process and methodology of the Strategy Server Event module for entry orders;

FIGURE 9 is a high-level flowchart depicting the automation process for strategy exit orders of the present invention;

20 FIGURE 10 is a flow chart that depicts the process and methodology of the Initial Checks module for exit orders;

FIGURE 11 is a flow chart that depicts the process and methodology of the Price Event from Data Server module for exit orders;

FIGURE 12 is a flow chart that depicts the process and methodology of the Place Trade Event module for exit orders;

FIGURE 13 is a flow chart that depicts the process and methodology of the Timer Event module for exit orders;

5 FIGURE 14 is a flow chart that depicts the process and methodology of the Trade Server Event module for exit orders;

FIGURE 15 is a flow chart that depicts the process and methodology of the Strategy Server Event module for exit orders;

10 FIGURE 16 is a flow chart that depicts the process and methodology of the Check for Multiple Entries module; and

FIGURE 17 is a flow chart that depicts the process and methodology of the Check for Multiple Exits module.

**DETAILED DESCRIPTION OF THE PREFERRED
EXEMPLARY EMBODIMENTS OF THE PRESENT INVENTION**

15 The following detailed description is presented to enable any person skilled in the art to make and use the invention. For purposes of explanation, specific nomenclature is set forth to provide a thorough understanding of the present invention. However, it will be apparent to one skilled in the art that these specific details are not required to practice the invention. Descriptions of specific applications
20 are provided only as representative examples. Various modifications to the preferred embodiments will be readily apparent to one skilled in the art, and the general principles defined herein may be applied to other embodiments and applications without departing from the spirit and scope of the invention. The present invention is

not intended to be limited to the embodiments shown, but is to be accorded the widest possible scope consistent with the principles and features disclosed herein.

With reference now to FIGURE 1 of the Drawings, there is illustrated therein a distributed computing network, generally designated by the reference numeral 100, utilizing the principles of the present invention. The various elements of the network are connected to each other via a network such as the Internet, indicated by the reference numeral 105. A computer, designated by reference numeral 110, represents a stand-alone user of the invention. It should be understood, however, that a user of the invention can instead be a wireless user 115, connecting to the network 105 via conventional telecommunications interfaces.

Other network elements include Pools of Liquidity 120 which may include Electronic Communication Networks (ECNs), Alternate Trading Systems (ATS), Market Makers and New York Stock Exchange Specialists. It should also be understood that a variety of third party information sources, generally designated by the reference numeral 125, may also be employed in the gathering of news events, market data and other information. Naturally, further sources of information are the various security and commodity exchanges 130 located throughout the world. Finally, a server or server farm 135 for performing the various financial calculations and managing the various accounts pursuant to the teachings of the present invention is also shown.

In operation, the user at their computer (collectively, "the user 110") creates a trading strategy to use for trading securities on his or her computer or wireless device 115. A unique aspect of the present invention is that the user 110 may choose a

strategy that has already been created or the user may create his or her own new strategy. Additionally, the invention allows the user to write new trading strategies on their computer 110 or wireless device 115 in English-like phrases that an automation process translates into computer executable code.

5 The user 110 then applies the trading strategy to real, historical data and back tests the strategy. By back testing the strategy, the user can check to see if the strategy would have been successful when applied to the historical market data. The data for the back testing is preferably stored on the server farm 135 for simplicity. It should, of course, be understood, however, that the data may be dispersed across a
10 number of computers or servers, whether within the server farm 135 or among more disparate and dispersed sources. In any event, the user 110 requests the data for a particular range of dates and symbols, and applies the trading strategy to the requested back data. Pursuant to the present invention, the user is then informed whether or not the trading strategy would have been successful on that historical data.

15 If the back test proves to be successful, the user 110 may then decide to apply the trading strategy to real-time market data and further test the strategy. The methodology of the present invention then automatically applies the rules of the trading strategy to the real-time market data, as discussed further hereinafter. The market data may come from the various stock exchanges 130, third party information
20 sources 125 or Pools of Liquidity 120. When the restrictions of a strategy are fulfilled, the user 110 is then prompted. For example, if a particular trading strategy is based upon a price restriction, the strategy may call for placing a buy order for XYZ stock at \$50. As soon as that restriction is fulfilled, the user 110 is alerted that

the restrictions of the rule have been fulfilled. After testing the trading strategy on real-time market data, the user 110 may then decide to execute the strategy to perform real trades in the market.

A further unique aspect of the present invention is the ability to monitor factors that affect entry and exit orders and trading strategies and to automatically modify entry and exit orders to ensure that the users' orders are properly executed. Depending on changing market factors, the methodology of the present invention will automatically alter, cancel or change the status of entry and exit orders. For example, a user's trading strategy may be based upon a price restriction that calls for buying ABC stock when the price falls to \$25. Thus, as soon as ABC falls to \$25, the user is alerted that the restriction has been fulfilled and prompted to buy the security.

As soon as the user sends an entry or exit order into the network 105, the methodology of the present invention begins to continually monitor the status of the order. Oftentimes an entry or exit order is not filled for a variety of reasons. By way of example, by the time an order reaches the appropriate Pool of Liquidity 120, the price restriction may no longer be fulfilled. For instance, if a user placed an order to purchase ABC at \$25 and by the time the order reached the Pools of Liquidity 120 the stock is no longer available at that price, the order is modified accordingly pursuant to the methodology of the instant invention. It should, of course, be well understood that it is not unusual for such occurrences to occur in a rapid trading environment. The system and methodology of the present invention preferably monitors all existing entry and exit orders and cancels or modifies orders so that all entry and exit orders are properly executed.

It should further be understood that the system and method of the present invention not only monitor external market conditions that occur at the stock exchanges 130 and Pools of Liquidity 120, but also monitor the trading strategies that are located on the various user devices, *i.e.*, computers 110 or wireless devices 115.

5 For example, a user may have an existing order to purchase 1,000 shares of XYZ stock that is sent to the market. In the meantime, the strategy that prompted the order might cancel or it might send an exit order or the user may close the strategy. The improved system and methodology of the instant invention preferably recognizes the strategy change, and modifies the order accordingly. For example, the case of a
10 cancel, the order is pulled out of the market very quickly. In the case of the strategy generating an exit, a check is made whether or not the market has filled the entry. If the order is not filled, the order is canceled. If, however, the market has filled the entry, an exit order is placed. Further, if the user deletes or closes the strategy, the user is alerted that there is an existing order in the market.

15 A further unique aspect of the present invention is the ability to automatically execute trading strategies into entry and exit orders. As discussed above, a user may create a trading strategy or select a pre-defined strategy pursuant to the teachings of the present invention. Once the user has back tested the strategy and has begun to place trades in the markets with the strategy, the user may automate the process of
20 monitoring the markets, strategies and orders for automatic placement of entry and exit orders based upon the users' trading strategy. For example, the user may create a simple strategy that buys ABC stock at \$40 and sells the stock at \$50. The user may automate the methodology to perform those trades without any direct action by the

user, *i.e.*, automatically buy ABC stock when it hits \$40 and sell the stock when it reaches \$50. It should, therefore, be understood that the system and methodology of the present invention are capable of fully automating the trading strategy once activated by the user. Alternatively, the user may not automatically place the trades, but instead may merely alert the user that the restrictions for a particular trading strategy are currently fulfilled and that there currently exists a trading opportunity.

Even though the above described examples are relatively simple strategies based upon elementary principles of buying and selling stocks for a profit, it is to be appreciated that the system and methodology of the present invention are capable of automating complex trading strategies that may be based upon intricate calculations of a number of different variables. Various inventive algorithms of the aforementioned automation processes of the present invention are discussed and explained in further detail in the following drawings and detailed description.

It should, of course, be understood that the automation process of generating and executing orders generates both entry orders and exit orders. Entry orders are generally buy or short orders, while exit orders are generally sell or cover orders. FIGURES 2-8 and accompanying text of the detailed description provide teachings and explanation of the automated process for entry orders. Similarly, FIGURES 9-15 and the accompanying text hereinbelow provide teachings and explanation of the automated process for exit orders generated by the automation process of the present invention.

With reference now to FIGURE 2 of the Drawings, there is illustrated a high-level flowchart depicting the automation process for strategy entry orders according to

an embodiment of the present invention. It should be understood to one skilled in the art that each element of the flowchart represents a different module of the automation process, generally designated by the reference numeral 200, for entry orders that is described in more detail in the following drawings and detailed descriptions. The automation process 200 begins when a strategy entry order is triggered (step 205). Next, initial checks are performed on the order (step 210). Initial Checks 210 determines whether or not the order has any time, position, or price restrictions since a restricted order cannot be sent to the market. The automation process 200 described herein preferably holds the order until the restrictions are lifted and the order can be placed or the order expires. If the order is not restricted, the order will immediately be sent to the trade server, *e.g.*, server 135 in FIGURE 1, to be filled in the market. After Initial Checks 210, the order is added to an Event Monitor 220, where the order waits to be filled by changing market conditions.

With reference again to FIGURE 2, there are illustrated various modules of the automation process, including a Price Event From Data Server 225, a Place Trade Event From User 230, a Timer Event 235, a Trade Server Event 240 and a Strategy Server Event 245. Price Event From Data Server 225, for example, receives all new orders from the Event Monitor 220 and determines if the orders can be filled in light of existing market conditions. The Place Trade Event From User 230 is a module of the automation process that performs internal checks on existing orders and determines if a restricted order has become unrestricted and can, therefore, be sent to be filled in the market. The Timer Event 235 is a module of the automation process that constantly monitors orders that have been placed into the market and alerts the

users if their order is not yet filled. The Trade Server Event 240 is another module of the automation process that monitors orders once they have been placed into the market. For example, it is not unusual for an entry or exit order to only be partially completed by a Pool of Liquidity 120. The Trade Server Event 240 will alert the user of such an occurrence and modify the order in accordance with the trading strategy. Strategy Server Event 245 is a module of the automation process that monitors the trading strategies on the users' computer. Strategy Server Event 245 determines whether or not a particular trading strategy is still in use and whether the entry orders that were triggered by that trading strategy should be canceled or removed.

It is to be appreciated that an order goes through many different stages during its existence. When an order is generated by the trading strategy it is in a sending status. Once the trading server receives the order it is in a sent status. When a Pool of Liquidity 120 receives the order, it is in the received status. Various states may result after the Pool of Liquidity receives the order. The order may be filled and the order is then referred to as being in the filled state. It may be partially filled and the order is then referred to as being in a partially filled state. The trader may cancel the order and receive a UROut confirmation from the Pool of Liquidity 120. A UROut may also be in the form of a Partial UROut confirmation that is the result of a partially filled order. The order may be rejected by the Pool of Liquidity which would result in a rejected state. Similarly, an order may be in a canceled state if the trader or strategy canceled the order prior to it being filled. And, the order may also be described as being in an expired state if the Pool of Liquidity 120 did not fill the order before the market closed.

With reference now to FIGURES 3A and 3B, there is illustrated a flow chart that depicts the process and methodology of the Initial Checks module 210 in FIGURE 2 in more detail. The algorithm for the Initial Checks module 210, generally designated in FIGURES 3A and 3B by the reference numeral 300, begins (step 301) where a strategy order to buy or short is triggered. The order then passes to an Initialize Flags routine (step 302), where various flags, that help monitor the order throughout the automation process, are set to their default value. As is understood to one skilled in the art, the flags indicate whether or not a particular order is restricted from being sent to the market. For example, a trader may put in an order to buy XYZ stock once it drops to \$50. The system and methodology of the present invention recognizes this restriction and sets a Price Restriction flag to TRUE. Once the price of XYZ drops to \$50, the price restriction is removed and the Price Restriction flag is set to FALSE. If a particular order has any restrictions, it will not be sent to the market and, therefore, will not be filled.

As discussed, the Initialize Flags 302 routine initializes the flags to their default settings. A Trade Server Status variable is, for example, initialized to Unsent. The phrase "Trade Server Status" describes the status of an order, and can be one of the following values: unsent (the order has not been sent from the user to the Trade Server), sending (the order is sent to Trade Server but an acknowledgement is not yet received), sent (the Trade Server has received the order), received (the Trade Server has routed the order to the appropriate market participant (Pool of Liquidity, Market Maker, NYSE Specialist, etc.)), filled (the market participant has totally filled the order), partially filled (Alive) (the market participant has partially filled the order and

the order is still active) partially filled (UROut) (the market participant has partially filled the order but the order is no longer active), cancel pending (the user has cancelled the order but the order is still active), UROut (the user has cancelled the order and the order is no longer active), TooLateToCancel/Cancel, Rejected (the user has cancelled the order but the order was already filled), expired (the order is no longer active due to time limitations), and rejected (the Pool of Liquidity was unable to accept the order). Each of the above possible status settings will be analyzed in more detail hereinbelow.

A Strategy Status flag is initially set to unfilled. The Strategy Status is the state of a strategy order within the methodology of the present invention, and can be one of the following values: active (the initial state of an order that is generated by a trading strategy), filled (the state of a strategy order when it is determined that the market would have filled the order), cancelled (the state of an order when the strategy has determined that the order is no longer valid) and deleted (the state of an order when the strategy that generated this order is no longer running).

A Price Restricted flag is initially set to FALSE. The Price Restricted flag is preferably a Boolean flag that is used to hold an order until a certain price is reached. It is used to implement stop orders. Likewise, a Time Restricted Flag is a Boolean flag initially set to FALSE, and is used to hold an order until a certain time is reached. The Time Restricted Flag is for orders that are generated during non-trading hours that should be placed when the market opens. A Position Restricted Flag is another Boolean flag initially set to FALSE, which is used to hold an order

until a flat (neutral) position is reached. For example, the Position Restricted Flag may be used to hold long entry orders from being placed if a short position exists.

A Multiple Exit Restricted flag is still another Boolean flag initially set to FALSE and is used to hold an exit order that would otherwise result in a greater exit
5 quantity than entry quantity. A Multiple Entry Restricted flag is a Boolean flag initially set to FALSE that is used to hold an entry (long or short) order that, if placed and filled, would result in a box position. Box positions occur when there are both long and short positions in a security simultaneously.

A Pending UROut flag is a Boolean flag initially set to FALSE and is used to
10 hold an exit (sell or cover) order while waiting for another previously placed exit order to be cancelled. A Pending Position Update flag is another Boolean flag initially set to FALSE that the invention uses to treat filled orders as open orders until the position has updated. The Pending Position Update flag is necessary since the methodology of the instant invention does not simultaneously notify the user that an
15 order has been filled and that there is new position information.

A Force to Limit flag is a Boolean flag initially set to FALSE and is used to notify the user that an entry or exit order has been generated but entry or exit orders are no longer accepted, and the order can only be placed as a limit order. Finally, a Warn if Unfilled flag is yet another Boolean flag initially set to FALSE and is used to
20 notify the user that a strategy order has been filled by the strategy but has not yet been filled by the market. It should, of course, be understood to one skilled in the art that the various flags enumerated above, although preferably of Boolean type, may be of

another type, *e.g.*, integer, which corresponds to a true and false setting, *e.g.*, zero and non-zero.

Returning now to the methodology illustrated in FIGURES 2, 3A and 3B, the Initial Checks Module 210 determines if there are any time restrictions on the order that was generated by the trading strategy, *e.g.*, by perusal of the status of the Time Restricted Flag. The order is then passed from Initialize Flags 302 to a current trading decision node 304, where it is determined whether or not the order was triggered in the current regular trading session. If the order was triggered in the current regular trading session, control is passed to a Check Entry Order Action 316, described in more detail hereinbelow. If, however, the order was not triggered in the current regular trading session, control then passes to an extended trading decision node 306, where it is determined if the order was triggered in a current extended trading session. If the order was not triggered in the current extended trading session, the aforementioned Time Restricted Flag is set to TRUE (step 308), and the order is passed to the Check Entry Order Action 316. If, however, the order was triggered during the extended trading session, the order is passed to an On Close Entry decision node 310, where it is determined if the order is an On Close Entry Order.

“On close” orders are orders that are executed at the close of a bar or the close of the market at the end of the day. If the order closes on the last bar of the day, then the order can no longer be placed in the regular market and the user is informed that that the order can only be placed in the extended session. If the order is an On Close Entry Order, then a Boolean flag for Force to Limit is set equal to TRUE (step 312), and the order is then passed to the aforementioned Check Order Entry Action 316. If,

however, the order is not an On Close Entry Order, the Time Restricted Flag is set to TRUE (step 314), and the order is also passed to the Check Entry Order Action 316.

With reference now to FIGURES 2 and 3B, the Initial Checks Module 210 next determines if there are any position restrictions on the order that was generated by the trading strategy. In particular, the Check Entry Order action 316 determines whether the order is a buy order or a short order. If it is determined that the order is a buy order, the order is passed to a short decision node 318, where it is determined if there is a short position or an open short order for the stock symbol in the account manager. If there is not a short position or an open short order for the stock symbol in the account manager, the aforementioned Position Restricted Flag is set to FALSE (step 320) and the order is passed to a Stop Order 328. An “open order” is defined as having a Trade Server State of “sending”, “sent”, “received”, “cancel pending” or “filled/partially filled (alive)” with a Pending Position Update set equal to TRUE. If there is a short position or an open short order for the stock symbol in the account manager (decision node 318), the order is passed to a short exit decision node 322, where it is determined whether there are short exit orders for the strategy with enough quantity to close the short position. If there are short exit orders for the strategy with enough quantity to close the short position, then the Position Restricted Flag is set to TRUE (step 324) and the order is then passed to the aforementioned Stop Order 328. If, however, there are no short exit orders for the strategy with enough quantity to close the short position, the user is warned (step 326) that they must manually exit the short position before the long entry order generated by the strategy will be placed.

The Position Restricted Flag is then set to TRUE (step 324), and the order is passed to the Stop Order 328.

Returning now back to the aforementioned Check Entry Order Action 316, if Check Entry Order Action 316 determines that the order is a short order, the order is then passed to a long decision node 330, where it is determined if there is a long position or an open long order for this symbol in the account manager. If there is not a long position or an open long order for this symbol in the account manager, the Position Restriction Flag is set to FALSE (step 320), and the order is passed to the Stop Order 328, as described hereinabove. If it is determined there is a long position or an open long order for this symbol in the account manager, then the order is passed to a long exit decision node 332, where it is determined whether or not there are any existing long exit orders for the strategy with enough quantity to close the long position. If there are existing long exit orders for the strategy with enough quantity to close the long position, the Position Restricted Flag is set to TRUE (step 336), and the order is passed to the Stop Order 328. If there are no existing long exit orders for the strategy with enough quantity to close the long position, the user is warned that they must manually exit the long position before the short entry order generated by the strategy will be placed (step 334). The Position Restricted Flag is then set to TRUE (step 336), and the order is passed to the Stop Order 328.

Next, the Initial Checks Module 210 determines if there are any price restrictions on the order that was generated by the trading strategy. Stop Order 328 determines if the order is a stop order. If the order is a stop order, the buy price restriction is set to execute the trade if the trading price is less than the stop price, or

the short price restriction is set to execute the trade if the trading price is greater than the stop price (step 338). The order is then passed to an order tab node 340, where the order is added to the account manager and the Check for Multiple Entries Module is called. The process of checking for multiple entries is described in more detail hereinbelow. If, however, Stop Order 328 determines that the order is not a stop order, then the order is passed directly to the aforementioned order tab 340, where the order is added to the account manager and the order is checked for multiple entries. The order then passes to the aforementioned Event Monitor 220, discussed in connection with FIGURE 2.

With reference now to FIGURE 4, there is illustrated a flow chart depicting the process and methodology of the Price Event from Data Server module 225 illustrated in FIGURE 2. The methodology of the Price Event from Data Server module 225, generally designated in FIGURE 4 by the reference numeral 400, starts where an order to buy or short is received from the Event Monitor 220. The Price Event from Data Server module 225 first determines if the order is time restricted (decision node 405). If the order is time restricted, the order is returned to the Event Monitor 220, as illustrated. If, however, the order is not time restricted, then the order is passed to an order type decision node 410, where the order type is determined. For example, if the order is a market or limit order, then the order is passed to a Check for Multiple Entries node 415. If, however, the order is a stop market or stop limit order, the order is instead passed to a price restriction decision node 420, where it is determined if the order is price restricted. If the order is not price restricted, then the order is passed to the aforementioned Check for Multiple Entries node 415. If,

however, the order is price restricted, the order is passed to an order action node 425, where the order action is determined. If the order action is to short, then the order is passed to a short node 430, where if the trade price is less than or equal to the stop price, the Price Restricted Flag is set equal to FALSE. If, however, the order action is to buy, then the order is passed to a buy node 435, where if the trade price is greater than or equal to the stop price, then the Price Restricted Flag is set equal to FALSE. The orders from both nodes 430 and 435 are then passed to the aforementioned Check for Multiple Entries node 415. All orders sent to the Check for Multiple Entries node 415 are then returned to the Event Monitor 220.

With reference now to FIGURE 5 of the Drawings, there is illustrated a flow chart depicting the process and methodology of the Place Trade Event module 230 illustrated in FIGURE 2. The methodology of the Place Trade Event module 230, general designated in FIGURE 5 by the reference numeral 500, starts where a non-restricted unsent order is received from the Event Monitor 220 at an order decision node 505, where it is determined whether there is an existing long order for a short entry or a short order for a long entry for the stock symbol. If it is determined that there is an existing long order for a short entry or a short order for a long entry for the stock symbol, control is passed to a short-long node 510, where the Position Restricted Flag is set to TRUE. The order is then returned to the aforescribed Event Monitor 220. If, however, it is determined that there is not an existing long order for a short entry or a short order for a long entry for the stock symbol in the account manager, then the order is passed to position decision node 515, where it is determined whether there is an existing long position for a short entry or a short

position for a long entry for the symbol. If there is an existing long position for a short entry or a short position for a long entry for the symbol in the account manager, control is passed to an open exit decision node 520, where it is determined if there is an open exit order that that would close the position. If there is not an open exit order that that would close the position, the user is warned (step 525) that they must manually exit the long/short position before the short/long entry order generated by the strategy will be placed. The Position Restricted Flag is then set equal to TRUE (step 510) and the order is returned to the Event Monitor 220. Returning now to the position decision node 515, if it is determined that there is not an existing long position for a short entry or a short position for a long entry for the symbol in the account manager, the user is asked to either place or ignore the order (step 530). If the user indicates order placement, the order is placed (step 535), where the order is sent to the trade server. If the user ignores the order (step 540), the order's Trade Server Status is set to ignored and control is returned to the Event Monitor 220.

With reference now to FIGURE 6, there is illustrated a flow chart that depicts the process and methodology of the Timer Event module 235 illustrated in FIGURE 2. The methodology of the Timer Event module 235, generally designated in FIGURE 6 by the reference numeral 600, constantly monitors orders that have been placed into the market and alerts the users if their order is not yet executed by the Pool of Liquidity 120. The Timer Event module methodology 600 begins by checking (step 605) if 1) the Strategy Status flag is filled, and 2) the Trade Server Status is unfilled, and 3) the current time (strategy filled time) is greater than a predetermined amount of time, and 4) the Warn if Unfilled Flag is TRUE, and 5) it is

during market hours, and 6) the order is not ignored. If the above criteria are all TRUE, then the entry order's Warn if Unfilled Flag is set to FALSE (step 610) and the user is then warned that a predetermined amount of time has passed since notification to place the order and that the strategy has filled the entry order, but the
5 order has not been filled by the market (step 615).

Control is then passed to a time decision node 620, where it is determined whether or not the current time is in a regular trading session. If the current time is in a regular trading session, the order's Time Restriction Flag is set to FALSE (step 625) and the order is returned to the Event Monitor 220. If the current time is not in a
10 regular trading session (step 620), it is determined whether the order is unsent and filled by the strategy (step 630). If the order is unsent and filled by the strategy, the user is warned (step 635) that the strategy has filled the order, but the market is now closed and the order is unsent and the order will be set to ignored. Further, the user will be given the option to place the order in the extended trading session, as
15 discussed hereinabove. The order is then returned to the Event Monitor 220. Returning now to step 630, if the order is not unsent and filled by the strategy, it is returned to the Event Monitor 220. Returning to the perform checks decision node 605, if any of the aforementioned criteria are FALSE, control transfers to the time decision node 620, and continuing thereafter as described hereinabove.

20 With reference now to FIGURES 7A and 7B, there is illustrated a flow chart that depicts the process and methodology of the Trade Server Event module 240 illustrated in FIGURE 2. The methodology of the Trade Server Event module 240, generally designated by the reference numeral 700, monitors entry orders once they

are placed into the market stream. The Trade Server Event module methodology 700 begins by determining whether the event is a position changed event for the order (step 702). A "position changed event" occurs when a trader's position for a stock changes. For example, a user of the present invention may have a certain position in a particular stock symbol and orders for that same symbol that are restricted because of their position. A position change event occurs when the restrictions on those orders may be lifted due to a change in the position in the symbol. If the incoming event is a position changed event, the Pending Position Flag is then set to FALSE (step 704). Next, matching exit orders are referenced (step 706) and the order's Position Restricted Flag is set to FALSE (step 708) before returning the order to the Event Monitor 220, as illustrated.

Returning back to step 702, if it is determined that the order received from the Event Monitor 220 is not a position changed event, the order is instead passed to an order status update node 710 where the entry's Trade Order Status is updated. The order is then passed to a server status decision node 712, where the invention checks the entry order's Trade Server Status. If, for example, the order's Trade Server Status is "sent," control is passed to an update node 714, where the methodology of the present invention updates the trade server sequence number before returning control back to the Event Monitor 220, as illustrated. If the order's Trade Server Status is "received," control is instead passed to a cancel node 716, which sends any cancel orders that were queued up during the sending state.

Returning again to the server status decision node 712, if it is determined that the order's Trade Server Status is "expired," control is passed to an exit decision node

718, where it is determined whether a corresponding exit order was filled by the strategy. If it is determined that a corresponding exit order was not filled by the strategy, control is returned to the Event Monitor 220, as illustrated in FIGURE 7. If, however, it is determined that a corresponding exit order was filled by the strategy, the user is warned (step 720) that the strategy has filled the order but the market is closed and the order is expired. The user is then given the option of placing the order in the extended trading session. Next, the order is time restricted and order's state is changed to unsent (step 722) before returning the order to the Event Monitor 220.

With reference now to FIGURE 7B and returning again to the server status decision node 712, if it is determined that the order's Trade Server Status is "rejected", "cancel rejected" or "too late", the user is warned (step 724) that a strategy order that he placed has been rejected or is too late before the order is passed to an exit order decision node 726, where matching exit orders are referenced. If no matching exit orders are found, the order is returned to the Event Monitor 220, as illustrated. If, however, matching exit orders are found, the order is passed to an order status decision node 728, where the status of the order is determined. If the order status is UROut and UROut Pending is set equal to TRUE, the exit order is removed (step 730) before returning the order to the Event Monitor 220. If, however, it is determined at the order status decision node 728 that the order status is filled, canceled, rejected, too late or partial fill (UROut), then exit orders with UROut Pending set equal to TRUE will be set equal to FALSE (step 732) before returning the order to the Event Monitor 220.

With further reference to the server status decision node 712, if it is determined that the order's Trade Server Status is "filled" or "partially filled", then the Pending Position update flag is set to TRUE (step 734) before the order is passed to the aforescribed exit order decision node 726, where matching exit orders are referenced. As discussed hereinabove, if no matching exit orders are found, the order is returned to the Event Monitor 220. If, however, matching exit orders are found, the order is passed to the order status decision node 728, where the automation process for the Trade Server Event module continues as described above.

Returning again to the server status decision node 712, if it is determined that the status of the entry order is "UROut", then the Position Restricted Flag is set to FALSE (step 736) if there is a short or long entry position and no open short or long entry orders exist, the current position is flat and there exists restricted long or short entry orders. The order is then passed to the exit order decision node 726, where matching exit orders are referenced, as described above. If no matching exit orders are found, then the order is returned to the Event Monitor 220. If matching exit orders are found, the order is passed to step 728, where the automation process for the Trade Server Event module continues as described above.

With reference again to the server status decision node 712 in FIGURE 7B, if it is determined that the status of the entry order is "partially filled (UROut)", then the order is then passed to the aforescribed exit order decision node 726, where matching exit orders are referenced. If no matching exit orders are found, then the order is returned to the Event Monitor 220. If matching exit orders are found, the

order is passed to step 728, where the automation process for the Trade Server Event module continues as described above.

With final reference to the server status decision node 712 in FIGURE 7A, if it is determined that the status of the entry order is not one of the enumerated defaults mentioned above, then the status is recognized as “other” and the order is returned to the Event Monitor 220, as illustrated (step 738).

With reference now to FIGURE 8A and FIGURE 8B of the Drawings, there is illustrated a flow chart that depicts the process and methodology of the Strategy Server Event module 245 illustrated in FIGURE 2. The methodology of the Strategy Server Event module, generally designated by the reference numeral 800, monitors the trading strategies on the users’ computer or wireless device, determines whether or not a particular trading strategy is still in use, and determines whether the entry orders that were triggered by the trading strategy should be canceled or removed.

The methodology of the Strategy Server Event module begins with a check of the entry order’s Strategy Server Status (step 802). If the entry order has been filled by the strategy, the entry order’s Strategy Status is set to filled and the Strategy Filled Time is set to the current time (step 804). Control is then returned to the Event Monitor 220, as illustrated in FIGURE 8A.

With reference again to the status check (step 802), if it is determined that the order has been cancelled or deleted by the strategy, then the order is passed to an order status decision node 806, where the order status is checked in the trade server. If it is determined that the status of the order is “sending”, “sent”, or “received”, control is passed to a cancellation decision node 808, as illustrated in FIGURE 8B,

where it is determined whether the order was deleted or canceled. If the order was “canceled”, control passes to a cancellation node 810, where the methodology of the present invention removes the entry order, adds a cancel order, or queues a cancel order, as is understood to those of skill in the art. If, however, it is determined that the order in cancellation decision node 808 was deleted, the user is warned that a strategy that they were tracking is no longer running and that there is an open order (step 812) and control is transferred to a second cancellation decision node 814. If at the aforementioned second cancellation decision node 814 the user cancels the order, control is passed to the aforementioned cancellation node 810, where the methodology of the present invention removes the entry order, adds a cancel order, or queues a cancel order, as discussed. If the user does not cancel the order at the second cancellation decision node 814, the entry order is removed (step 816).

With reference again to the order status decision node 806 in FIGURE 8B, if it is determined that the entry order status is partially filled, control passes to a position decision node 818, where it is determined whether there is a position from the entry for that particular symbol. If there is no position from the entry for that particular symbol, then the user is warned (step 812) that a strategy that they were tracking is no longer running and that there is an open order. If the user thereafter cancels the order at the second cancellation decision node 814, control passes to the aforementioned cancellation node 810, where the automation process removes the entry order, adds a cancel order, or queues a cancel order. If, however, the user does not cancel the order, control instead passes to step 816, where the automation process removes the entry order, as discussed hereinabove.

Returning now to the position decision node 818, if it is determined that there is a position from the entry for that particular symbol, the user is warned (step 820) that a strategy they were tracking is no longer running or is cancelled but there is an open order that has been partially filled. If the user thereafter at an action decision
5 node 822 decides to cancel the entry order and close the position, the entry order is so removed, and a cancel and close position order is added (step 824). If the user instead decides to do nothing, the entry order is just removed (step 826). Alternatively, if the user decides to just cancel the entry order, the entry order is removed and a cancel order is added (step 828).

10 With reference again to the order status decision node 806, if it is determined that the order status is filled or partial fill (UROut), control is passed to a position decision node 830, as illustrated in FIGURE 8A, where it is determined if there is a position from this entry for that particular symbol. If there is no position from this entry for that particular symbol, then the entry order is removed (step 832). If,
15 however, there is a position from this entry for that particular symbol the user is warned (step 834) that a strategy that they were tracking is no longer running or is cancelled but a position has been taken. If the user thereafter at an action decision node 836 decides to close the position, the entry order is removed and an order to close the position is added (step 838). If, however, the user decides not to close the
20 position, the entry order is removed (step 832), as discussed hereinabove.

With final reference to the order status decision node, if it is determined that the order's status in the trade server is a status not mentioned above, the entry order is removed (step 840), as illustrated in FIGURE 8B.

With reference now to FIGURE 9 of the Drawings, there is illustrated a high-level flowchart depicting the automation process for strategy exit orders, generally designated by reference numeral 900, according to an embodiment of the present invention. The automation process for the exit orders is similar to the process for the entry orders as described in FIGURE 2. As with the entry order automation process as described in FIGURE 2, the exit order automation process 900 begins when a strategy exit order is triggered (step 905). Next, initial checks are performed on the order (step 910). Initial Checks 910 determines whether or not the exit order has any time, position, or price restrictions. Next, the order is added to an Event Monitor 220, where the order waits to be filled by changing market conditions. With reference again to FIGURE 9, there are illustrated various modules of the automation process, including a Price Event From Data Server 925, a Place Trade Event From User 930, a Timer Event 935, a Trade Server Event 940 and a Strategy Server Event 945. The aforementioned separate modules of the automation process for exit orders perform similar functions as the corresponding modules in the entry order automation process described in FIGURE 2.

With reference now to FIGURES 10A and 10B, there is illustrated a flow chart that depicts the process and methodology of the Initial Checks module 910 in FIGURE 9 in more detail. The algorithm for the Initial Checks module 910, generally designated in FIGURES 10A and 10B by the reference numeral 1000 begins when an exit strategy order is triggered. The order then passes to an Initialize Flags routine (step 1002) where various flags, that help monitor the exit order throughout the automation process, are set to their default value. As with the entry order automation

process flags, the exit order automation process flags are used to indicate whether or not a particular order is restricted from being sent to the market. The automation process utilizes the same flags for exit orders as are used for entry orders, as described in FIGURE 3. Further, the flags utilized for exit orders are set to the same default settings as the corresponding entry orders flags described above in FIGURE 3.

Next, Initial Checks Module 910 determines if there are any Time Restrictions on the order that was generated by the trading strategy, *e.g.*, by perusal of the status of the Time Restricted Flags. The order is then passed from Initialize Flags 1002 to a current trading decision node 1004, where it is determined whether or not the order was triggered in the current regular trading session. If the order was triggered in the current regular trading session, control is passed to a Check for Matching Entry Order 1016 described in more detail hereinbelow. If, however, the order was not triggered in the current regular trading session, control then passes to an extended trading decision node 1006 where it is determined if the order was triggered in a current extended trading session. If the order was not triggered in the current extended trading session, the Time Restricted Flag is set to TRUE (step 1014) and the order is passed to the Check for Matching Entry Order 1016. If, however, the order was triggered during the extended trading session, the order is passed to an On Close Entry decision node 1008 where it is determined if the order is an On Close Entry Order. If the order is an On Close Entry Order, then the Boolean Flag for Force to Limit is set equal to TRUE (step 1012) and the order is then passed to the aforementioned Check for Matching Entry Order 1016. If, however, the order is not

an On Close Entry Order, the Time Restricted Flag is set to TRUE (step 1010) and the order is passed to Check for Matching Entry Order 1016.

With reference now to FIGURE 10B, the Initial Checks Module 910 next determines if there are any position restrictions on the exit order that were generated by the trading strategy. In particular, the Check for Matching Entry Order 1016 checks for matching entry orders. If matching entries are found the order is passed to a filled decision node 1018 where it is determined if the matching entry order is filled or partially filled. If the order is not filled or partially filled, the aforementioned Position Restricted Flag is set equal to TRUE (step 1024) and the order is passed to a Stop Order 1030.

With reference again to Check for Matching Entry Order 1016, if the order is filled or partially filled the order is passed to a position decision node 1020 where if the order is a sell order, long positions are referenced and if the order is to cover, short positions are referenced. If, at decision node 1020, the described positions are found, the order is passed to 1022 where all open exit orders for the matching entry are found and subtracted from the entry quantity. If at node 1022 the exit order's quantity is greater than the net quantity, the Pending Exit UROut flag is set equal to TRUE and the Position Restricted Flag is set equal to FALSE (step 1026) before the order is passed to a Stop Order 1030. Returning now to position decision node 1020, if the aforementioned positions are not found, the Position Restricted Flag is set equal to TRUE (step 1028) and the order is passed to Stop Order 1030.

Next, the Initial Checks module 910 determines if there are any price restrictions on the exit order that were generated by the trading strategy. Stop Order

1030 determines whether the order is a stop order. If the order is a stop order the cover price restriction is set to execute the trade if the trade price is less than the stop price, or the sell price restriction is set to execute the trade if the trade price is greater than the stop price (step 1032). The order is then passed to an Order Tab node 1034
5 where the order is checked for multiple entries. If Stop Order 1030 determines that the order is not a stop order then the order is passed to the Order Tab node 1034 where the order is added to the account manager and checked for multiple entries. The Order Tab node 1034 then passes the order to the aforementioned Event Monitor 220 discussed in connection with FIGURE 2.

10 With reference now to FIGURE 11, there is illustrated a flow chart depicting the process and methodology of the Price Event from Data Server module 925 illustrated in FIGURE 9. The methodology of the Price Event from Data Server module 925 generally designated in FIGURE 11 by reference numeral 1100, starts when an exit order is received from the Event Monitor 220. The Price Event from
15 Data Server module 225, first determines if the order is time restricted (decision node 1105). If the order is time restricted, the order is returned to the Event Monitor 220, as illustrated. If, however, the order is not time restricted, the order is passed to an order type decision node 1110 where the order type is determined. For example, if the order is a market or limit order, then the order is passed to a Check for Multiple
20 Entries (node 1120). If, however, the order is a stop market or stop limit order, the order is passed to a price restriction decision node 1115 where it is determined if the order is price restricted. If the order is not price restricted, then the order is passed to the aforementioned Check for Multiple Entries (node 1120). If the order is price

restricted, the order is passed on to order action node 1125 where the order action is determined. If the order action is to sell, then the order is passed to a sell node 1130, where if the trade price is less than or equal to the stop price, the Price Restricted flag is set equal to FALSE. If, however, the order action is to cover, the order is passed to
5 a cover node 1135, where if the trade price is greater than or equal to the stop price, the Price Restricted flag is set equal to FALSE. The orders from both nodes 1130 and 1135 are then passed to the aforementioned Check for Multiple node 1120. All orders sent to the Check for Multiple Entries mode 1120 are then returned to the Event Monitor 220.

10 With reference now to FIGURE 12 of the Drawings, there is illustrated a flow chart depicting the process and methodology of the Place Trade Event module 930 illustrated in FIGURE 9. The methodology of the Place Trade Event module, generally designated by the reference numeral 1200, starts when a non-restricted unsent order is received from the Event Monitor 220. At an order decision node
15 1205, the corresponding entry order is referenced. If the corresponding entry order is not found, the exit order is passed to position decision node 1210 where long positions are checked for sell orders and short positions are checked for cover orders. If the described positions are not found, the exit order's Position Restricted Flag is set equal to TRUE (step 1215) and the order is returned to the Event Monitor 220. If,
20 however, at position decision node 1210, the described positions are found, the user is warned that no matching entry orders exist but that a position does exist (step 1220). If, at exit decision node, the user decides not to exit the position, the exit order is set to ignored (step 1225) and is returned to the Event Monitor 220. If, at exit decision

node 1220, the user decides to exit the position, the order is passed to quantity node 1230, where the exit order quantity is reduced, if necessary. Next, the user is prompted to either place the exit order or ignore the exit order (step 1235). If the user places the exit order, the order is sent to the trade server (step 1240) before control is
5 returned to the Event Monitor 220. If, however, the user ignores the exit order, the exit order's Trade Server Status is set to ignored (step 1245) before the order is returned to the Event Monitor 220.

Returning now to order decision node 1205, if the entry order is found, the order is passed to leaves decision node 1250, where it is determined if the found
10 entries have leaves. Leaves refer to the quantity that remains after a partial fill. For example, if an order to buy 1000 shares of XYZ stock is partially filled with 600 shares, then there are 400 remaining shares to be filled. The 400 shares are referred to as leaves. If, at leaves decision node 1250, the found entries do not have leaves, the order is passed to quantity node 1230 where the order's quantity is reduced, if
15 necessary, and the order continues through the remainder of the Place Trade Event automation process 930 as described above. If, however, at order decision node 1250, it is determined that the found entries do have leaves, the order is passed to cancel decision node 1255 where the user is informed that the entry order that is being exited has only partially been filled. If, at cancel decision node 1255, the user decides to
20 cancel the outstanding entry order, the exit order's quantity is adjusted to reflect the entry order's filled quantity (step 1260) and the order is the passed to decision node 1235 where the order continues through the remainder of the Place Trade Event automation process 930 as described above. If at cancel decision node 1255, the user

decides not to cancel the outstanding entry order, the exit order's quantity is adjusted to reflect the entry order's filled quantity (step 1265) and a new exit order is added for the leaves with the Position Restricted Flag set equal to TRUE. The order is then passed to decision node 1235 where the order continues through the remainder of the

5 Place Trade Event automation process 930 as described above.

With reference now to FIGURE 13, there is illustrated a flow chart that depicts the process and methodology of the Timer Event module 935 illustrated in FIGURE 9. The methodology of the Timer Event module 935, generally designated in FIGURE 13 by the reference numeral 1300, constantly monitors exit orders that have
10 been placed into the market and alerts the users if their order is not yet executed by a Pool of Liquidity 120. The Timer Event module methodology 1300, begins by checking (step 1305) if 1) the Strategy Status Flag is filled, and 2) the Trade Server Status is unfilled, and 3) a prescribed period of time has elapsed and 4) the Warn if Unfilled Flag is TRUE, and 5) it is during market hours, and 6) the order is not
15 ignored. If the above criteria are all TRUE, then the exit order's Warn if Unfilled flag is set to FALSE (step 1310) and the user is then warned that a predetermined amount of time has passed since notification to place the order and that the strategy filled the entry order, but the order has not been filled by the market at (step 1315). Control is then passed to a time decision node 1320 where it is determined whether or not the
20 current time is in a regular trading session. If the current time is in a regular trading session, the order's Time Restriction Flag is set to FALSE (step 1325) and the order is returned to the Event Monitor 220. If the current time is not in a regular trading session (step 1320), it is determined whether the order is unsent and filled by the

strategy (step 1335). If the order is unsent and filled by the strategy, the user is warned (step 1330) that the strategy has filled the order, but the market is now closed and the order is unsent and the order will be set to ignored. Further, the user will be given the option to place the order in the extended trading session as discussed
5 hereinabove. The order is then returned to the Event Monitor 220. Returning now to step 1335, if the order is not unsent and filled by the strategy, it is returned to the Event Monitor 220. Returning to the perform checks decision node 1305, if any of the aforementioned criteria are FALSE, control transfers to the time decision node 1320, and continuing thereafter as described hereinabove.

10 With reference now to FIGURES 14A and 14B, there is illustrated a flow chart that depicts the process and methodology of the Trade Server Event module 940 in FIGURE 9. The methodology of the Trade Server Event module 940, generally designated by reference numeral 1400, monitors market exit orders once they are placed into the market stream. The Trade Server Event module methodology 940
15 begins by determining whether the event is a position changed event for the order (step 1402). If it is a position changed event, the pending position flag is set to FALSE (step 1429). Next, at restricted node 1428, it is determined if there are any entry orders for the symbol that are position restricted with a quantity of 0. If there are such orders, the Position Restricted Flag is set to FALSE (step 1430) and the order
20 is returned to the Event Monitor 220 as illustrated. If, however, at restricted node 1428, it is determined there are not any entry orders for the symbol that are position restricted with a quantity of 0, the order is then returned to Event Monitor 220, as illustrated.

Returning now to position changed node 1402, if it is determined that the Trade Server Event received from the Event Monitor 220 is not a position changed event, the trade order status is updated (step 1404). The order's Trade Server Status is then checked at server status decision node (step 1406). If the order's Trade Server Status is "sent," the automation process then updates the trade server sequence number before returning control back to the Event Monitor 220 (step 1408). If it is determined at server status decision node 1406 that the order's Trade Server Status is "received," the automation process will then send any cancel orders that were queued up during the sending state (step 1410) before returning control to the Event Monitor 220.

With reference to FIGURE 14B, server status decision node, designated by reference numeral 1406, represents the same server status decision node 1406 in FIGURE 14A. Returning now to server status decision node (step 1406), if it is determined that the order's Trade Server Status is "expired," it is then determined whether the exit order was filled by the strategy (step 1412). If it is determined that the order was not filled by the strategy, the order is returned to the Event Monitor 220. If it is determined that the exit order was filled by the strategy (step 1412), the user is warned (step 1414) that the strategy has filled the order but the market is closed and the order is expired. The user is then given the option of placing the order in the extended trading session (step 1414). Finally, the order is time restricted and the order's state is changed to unsent before returning the order to the Event Monitor 220 (step 1416).

With reference again to FIGURE 14A and returning to server status decision node 1406, if it is determined that the order's Trade Server Status is "rejected" or "too late," the user is warned that the strategy order has been rejected or is too late (step 1418) before the order is returned to the Event Monitor 220. Returning again to server status decision mode 1406, if it is determined that the order's Trade Server Status is "filled" or "partially filled," the Pending Position Update Flag is set to TRUE (step 1420) before the order is returned to the Event Monitor 220. Returning again to server status decision node 1406, if it is determined that the status of the entry order is "UROut" or "partial fill (UROut)" and the exit's quantity is greater than the net quantity, the order's Set Pending ExitUROut flag is set to TRUE before returning the order to the Event Monitor 220 (step 1422). If, however, it is determined that the status of the entry order is "UROut" or "partial fill (UROut)" and the exit's quantity is less than the net quantity, the order's Set Pending ExitUROut flag is set to FALSE before returning the order to the Event Monitor 220 (step 1422). Returning again to server status decision node 1406, if it is determined that the status of the exit order is not one of the statuses mentioned above, then the status is recognized as "other" and the order is returned to the Event Monitor 220 as illustrated in FIGURE 14A (step 1424).

With reference now to FIGURES 15A and 15B of the Drawings, there is illustrated a flow chart that depicts the process and methodology of the Strategy Server Event module for exit orders 945 illustrated in FIGURE 9. The methodology of the Strategy Server Event 945 is a module, generally designated by the reference numeral 1500, of the automation process that monitors the trading strategies on the

users' computer, and determines whether or not a particular trading strategy is still in use, and determines whether the market exit orders that were triggered by the trading strategy should be canceled or removed.

The methodology of the Strategy Server Event module begins with a check of the exit order's Strategy Server Status (step 1502) as illustrated in FIGURE 15A. If the exit order has been filled by the strategy, the exit order's Strategy Status is set to filled, and the Strategy Filled Time is set to the current time (step 1504). The entry order is then referenced (step 1506). If the entry order is not found, the order passes to position decision node 1518 where it is determined whether or not there is a position. If there is a position, the order is returned to the Event Monitor 220, as illustrated. If, however, there is not a position, the exit order is removed (step 1520).

Returning now to step 1506, if the entry order is found, the entry order status is determined at order status decision node 1508. If it is determined that the entry order status is "filled" or "partially filled (alive)," the order is returned to the Event Monitor 220 (step 1511). If, at order status decision node 1508, it is determined that the entry order status is "unsent," the exit order is removed and the entry order's quantity is reduced or the entry order is removed (step 1510). If, at order status decision node 1508, it is determined that the status of the entry order is "sending," "sent" or "received" a cancel order for the entry order is added and the Pending UROut flag is set to TRUE before the order is returned to the Event Monitor 220 (step 1512). If, at order status decision node 1508, the status of the order is "Partially Filled (UROut)," the exit order's quantity is reduced (step 1514) before the order is

returned to the Event Monitor 220. If, at order status decision node 1508, the status of the order is a status not mentioned above, the exit order is removed (step 1516).

Returning now to status check (step 1502), and with reference now to FIGURE 15B, if it is determined that the order has been cancelled or deleted by the
5 strategy, then the exit order status is checked in the trade server (step 1522). If, at step 1522, it is determined that the status of the order is “sending,” “sent,” or “received,” the canceled decision node 1524 determines whether the order was deleted or canceled. If the order was canceled, the automation process removes the exit order, adds a cancel order, or queues a cancel order (step 1526). If, however, at
10 canceled decision node 1524, it is determined that the order was deleted, the user is warned (step 1528) that a strategy being tracked is no longer running and that there is an open order. If the user cancels the order, the automation process removes the entry order, at warning decision mode 1528, adds a cancel order, or queues a cancel order (step 1526). If the user does not cancel the order, the automation process removes the
15 exit order (step 1530).

Returning now to step 1522, if it is determined that the entry order status is “filled” or “partially filled (UROut),” the exit order is removed (step 1534). Again, at step 1522, if it is determined that the status of the exit order is of another status, the exit order is removed at 1534 (step 1534).

20 With reference now to FIGURE 16 of the drawings, there is illustrated a flow chart that depicts the process and methodology of the Check for Multiple Entries module of the automation process generally designated by the reference numeral 1600. The Check for Multiple Entries module 1600 ensures that the automation

process does not simultaneously place conflicting orders. For example, if a trader has a long position for a particular symbol, the Check for Multiple Entries function will not allow a simultaneous entry order for a short position for the same symbol. The Check for Multiple Entries module 1600 determines which entry order is the closest
5 to the current trading price and restricts the other orders. The Check for Multiple Entries module further calculates a tolerance range that prevents minor fluctuations in the trading price from causing the automation process to repeatedly cancel and replace these orders.

The Check for Multiple Entries module begins when the module is called by
10 the automation process modules (step 1610). First, the Check for Multiple Entries module checks for active strategies with entries on both the long and short side that have a Trade Server Status of “unsent,” “sent,” “sending,” “received,” “partial fill (alive)” or “UROut” (step 1620). If a strategy is not found, the module returns a FALSE (step 1630).

15 If, however, a strategy is found with one of the above mentioned strategies, the Check for Multiple Entries module 1600, next finds the entry order that has the closest order price to the current trade price and that is not restricted or ignored (step 1640). If an order is not found, control passes to step 1630 where the Check for Multiple Entries module 1600 returns a false. If, however, the closest entry order is
20 for a long entry, the Set Restrictions node 1650 sets the Multiple Entry Restricted flag to FALSE for all other long entries for the strategy and sets the Multiple Entry Restricted flag to TRUE for all short entries for the strategy. Further, if an order was cancelled, the Multi Pending UROut flag is set to TRUE for all long entries in the

strategy mode the Multi Pending UROut restriction is removed when the UROut notice is received in the Trade Server Event for the cancelled order. Next, a new tolerance range is calculated (step 1660) before returning a TRUE (step 1670).

The tolerance range is used to prevent minor fluctuations in price from canceling and replacing orders. By way of example, there may be an order to buy at \$15 and an order to short at \$25 and the market price is currently fluctuating around \$20. Thus, every time the market goes above and below \$20, the automation process has to cancel either the buy or short order and replace it with the other. A tolerance level solves this problem by creating a zone where the automation process does not alter the active order. For example, if the tolerance zone is between \$19 and \$21 for the above example and the market is trading at \$20½, the market would have to drop below \$19 before the automation process would replace the short order with an active buy order for the symbol. Likewise, to cancel the buy order and replace it with a short order, the market would have to rise above \$21.

Returning now to Closest Price 1640, if the closest entry order is for a short entry, the Set Restrictions node 1645 sets the Multiple Entry Restricted flag to FALSE for all other short entries for the strategy and sets the Multiple Entry Restricted flag to TRUE for all long entries for the strategy. Further, if an order was cancelled, the Multi Pending UROut flag is set to TRUE for all short entries in the strategy and the Multi Pending UROut restriction is removed when a UROut notice is received in the Trade Server Event for the cancelled order (step 1645). Next, a new tolerance range for the orders is calculated (step 1660). Finally, the Check for Multiple Exits function then returns a TRUE (step 1670).

With reference now to FIGURE 17 of the Drawings, there is illustrated a flow chart that depicts the process and methodology of the Check for Multiple Exits module of the automation process generally designated by the reference numeral 1700. Like the Check for Multiple Entries module 1600, the Check for Multiple Exits module ensures that the automation process does not simultaneously place conflicting orders

The Check for Multiple Exits module begins when another module of the automation process calls the Check for Multiple Exits module 1700 (step 1705). The Check for Multiple Exits module 1700 checks for multiple exit orders for the strategy for a specific entry at Check Strategy Orders Tab 1710. The Check for Multiple Exits module 1700 only references active exit orders with a Trade Server Status of “unsent,” “sent,” “sending,” “received,” “partially filled,” or “UROut” (step 1710). If an exit order is not found, the module returns a FALSE (step 1715).

If, however, an exit order is found at Check Strategy Orders Tab 1710, the module sets the Multiple Exit Restricted flag to TRUE for each exit order (step 1720). Control then passes to Closest Price 1725 where the module determines which exit order has the closest order price to the current trade price, has the Multiple Exit Restricted Flag set to TRUE and isn’t otherwise restricted. The Closest Price 1725 then sets the Multiple Exit Restricted Flag to FALSE.

Control then passes to Quantity 1735 where it is determined if the quantity of the exit orders with Multiple Exit Restricted Flags set to FALSE is greater than or equal to the Position Quantity. If the quantity of the exit orders is greater than the position quantity, control passes to Tolerance Calculation 1740 where a new tolerance

range is calculated. If, however, at 1735, the quantity of the exit orders is not greater than the position quantity, control passes to All Orders decision node 1730 where it is determined if all of the exit orders have been checked. If all of the exit orders have not been checked, control passes back to Closest Price 1725. If, however, at All
5 Orders 1730, it is determined that all of the exit orders have been checked, control passes to Tolerance Calculation 1740 where a new tolerance range is calculated. The tolerance range for the Multiple Exits module 1700 functions much in the same way as in the Multiple Entries module 1600 described hereinabove.

Control then passes to Cancel 1745, where the Check For Multiple Exits
10 module 1700 cancels any exit orders that are in the market and have a Multiple Exit Restricted flag set to TRUE. If an order is cancelled, the set Multi Pending UROut flag is set to TRUE for all exit orders without any multiple exit restrictions. The Multi Pending UROut restriction is then removed when the UROut is received in the Trade Server Event for the cancelled order (step 1745). Finally, a TRUE is returned
15 at step 1750.

Although the system and methodology of the instant invention are at present best suited for a fixed or standalone computer, *e.g.*, the computer 110 in FIGURE 1, as wireless devices become more prevalent and bandwidth and capabilities increase, the principles of the present invention will become increasingly applicable to a
20 wireless trading environment via wireless devices, *e.g.*, the wireless device 115 depicted in FIGURE 1.

The foregoing description of the present invention provides illustration and description, but is not intended to be exhaustive or to limit the invention to the precise

one disclosed. Modifications and variations are possible consistent with the above teachings or may be acquired from practice of the invention. Thus, it is noted that the scope of the invention is defined by the claims and their equivalents.

11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221